



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

METODY ZVÝRAZŇUJÍCÍ DETAILS VE FOTOGRAFII

PHOTOGRAPHIC DETAIL ENHANCEMENT METHODS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PAVEL SEDLÁŘ

VEDOUcí PRÁCE

SUPERVISOR

Doc. Ing. MARTIN ČADÍK, Ph.D.

BRNO 2017

Abstrakt

Cílem této bakalářské práce, jež spadá do oblasti počítačového vidění, je seznámení se s problematikou metod umožňující zvýraznění detailu v obraze. Tato práce dále popisuje konkrétní metody, které budou posléze implementovány a se kterými se bude v neposlední řadě experimentovat pro další použití.

Abstract

The aim of this Bachelor Thesis, which falls within the field of computer vision, is to get acquainted with the issue of methods that can highlight detail in the image. This work also describes specific methods that will later be implemented and with which, last but not least, experiment for further use. (google/translator)

Klíčová slova

zpracování, obraz, detail, zvýraznění, c++, optimalizace, histogram, ekvalizace, interpolace

Keywords

processing, image, detail, enhancement, c++, optimalization, histogram, equalization, interpolation

Citace

SEDLÁŘ, Pavel. *Metody zvýrazňující detaily ve fotografii*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. Ing. Martin Čadík, Ph.D.

Metody zvýrazňující detaily ve fotografii

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Martina Čadíka Další informace mi poskytli... Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Pavel Sedlář
16. května 2018

Poděkování

Tímto bych chtěl poděkovat především panu Martinu Čadíkovi při odborných konzultacích, radách a otázkách, na které mi ochotně a věcně odpověděl.

Obsah

1	Úvod	3
2	Zvýraznění detailů	4
2.1	Hlavní cíle	5
2.2	Obecný postup	5
2.2.1	Předcházející úprava obrazu	5
2.2.2	Zvýraznění detailů v obraze	6
2.3	Výhody a nevýhody	6
2.4	Použití	7
2.5	Ukázka jednoduché metody - Ekvalizace histogramu	7
3	Popis vybraných metod - teoretická část	8
3.1	Art-Photographic Detail Enhancement	8
3.1.1	Stručný popis	9
3.1.2	Výhody a nevýhody	9
3.1.3	Použití	9
3.2	Contrast limited adaptive histogram equalization	9
3.2.1	Stručný popis	10
3.2.2	Výhody a nevýhody	10
3.2.3	Použití	11
4	Popis vybraných metod - praktická část	12
4.1	Art-Photographic Detail Enhancement	12
4.1.1	Model tónové transformace	12
4.1.2	Dekompozice základní a detailní vrstvy	13
4.1.3	Maximalizace detailů	15
4.2	Contrast limited adaptive histogram equalization	19
4.2.1	Yxy barevný model	19
4.2.2	Histogram equalization	20
4.2.3	Adaptive histogram equalization	23
4.2.4	Clip Limit	27
5	Porovnání metod	30
5.1	Rychlost	30
5.2	Paměťová náročnost	30
5.3	Efektivita	31
5.4	Použití	31

6 Závěr	32
Literatura	33
A Obsah přiloženého média	36
B Použité technologie a knihovny	37
B.1 OpenCV knihovna	37
B.2 CGAL knihovna	37

Kapitola 1

Úvod

Zpracování obrazu zahrnuje mnoho operací se vstupním obrazem. Mezi tyto operace patří například převod barevného obrazu do stupňů šedi, úprava jasů a kontrastu, jeho rozmazání či rozostření, redukce šumu a mnoho dalších operací. Toto odvětví se proto vyskytuje v mnoha oborech, ve kterých usnadňuje práci lidem. Mezi tyto obory můžeme řadit zdravotnictví (rentgenové snímky), zábavní průmysl (filmy, reklamy, hudební videoklipy atd.), umění (úprava fotografií, filtry, restaurování uměleckých děl), architekturu (zjištění stáří historických památek, analýza povrchu), strojírenský průmysl (vady na výrobcích, ověření jakosti), kriminalistické techniky (daktyloskopie, forenzní analýza, rozpoznání osob na záznamech z kamer), atd.

Jednou z operací pro zpracování obrazu je v neposlední řadě i zvýraznění detailů, které se vyskytují v obraze. Důvodů pro použití metod, jež provádějí zvýraznění detailů, může být mnoho. Mezi tyto důvody můžeme řadit některé z důvodů uvedených v předchozím odstavci, např. estetickou úpravu obrazu, nalezení vady na výrobku, úpravu rentgenového snímku aj. Tato bakalářská práce se zabývá právě touto problematikou na následujících stranách.

Bakalářská práce je rozdělena do více kapitol. V úvodní kapitole se práce zaměřuje na zvýraznění detailů v obraze obecně, je popsán obecný postup realizace, definovány hlavní cíle, popsány výhody či nevýhody a nakonec uplatnění v oborech. Další kapitola se zabývá teoretickým popisem vybraných dvou metod, popisem jejich operací a jejich konkrétními uplatněními a výhodami či nevýhodami. V předposlední kapitole se práce zaměřuje především na praktickou část metod, kdy jsou popsány jednotlivé algoritmy a jejich implementace do kódu. V poslední kapitole se objektivně obě metody porovnávají z hlediska jejich rychlosti, efektivity, paměťové náročnosti a způsobu použití.

Kapitola 2

Zvýraznění detailů

Kapitola se zabývá s obecnými principy zvýraznění detailů, hlavními cíli definující správné zvýraznění, obecným postupem, který by se měl uplatnit v metodách, výhody a nevýhody pramenící ze zvýraznění pomocí jednoduchých metod a oblastmi, ve kterých lze metody využít či jejich příklady použití. Jako vzor pro ukázkou použití ekvalizace histogramu byl použit nejspíš nejčastěji používaný obraz o velikosti 512x512 pixelů znázorňující švédskou dívku jménem Lena Söderberg, která v době pořízení snímku pózovala pro nejmenovaný pánský časopis.



Obrázek 2.1: Lena - Původní obrázek. Zdroj obrázku [11]

V této kapitole a jejích podkapitolách je popsán obecný postup, který metody pro zvýraznění detailů v obraze používají. Jsou zde popsány nejčastější hlavní cíle, kterých se snaží metody zpracovávající obraz dosáhnout, popsán obecný postup, jaké jsou výhody a nevýhody vzniklé z použití těchto metod ke zpracování obrazu a na závěr ukáзка jednoduché metody.

2.1 Hlavní cíle

Každá metoda pro zvýraznění detailů je odlišná a reprezentuje postup zvýraznění detailů v obraze jinou (nebo více) technikou (nebo technikami) nebo se používá na určitý druh digitálního obrazu, proto existuje více cílů, které jsou vytyčeny pro správnou implementaci těchto metod. Hlavním cílem metod zvýrazňujících detaily v obraze je zajistit zvýšení množství informací, který může upravený obraz poskytnout oproti původnímu obrazu. Dále se za cíl použití těchto metod může považovat eliminace šumu, který je způsoben například elektromagnetickým zářením při snímání fotoaparátem. Když se shrnou předchozí dvě věty dohromady, vznikne jeden cíl, při kterém by metoda samozřejmě měla poznat, jestli se na pixelu nachází detail, nebo se jedná pouze o šum či jiný nežádoucí artefakt, což je velice problematické na realizaci. V souvislosti pořízení snímku pomocí fotoaparátu by se mohl za cíl považovat také zostření rozmazaného obrazu. Dále jsou tyto metody velmi náročné na paměť, protože ke zpracování obrazů používají mnoho informací, proto je nutné použít při jejich implementacích takových technik, aby se riziko plné paměti co nejvíce eliminovalo. Způsobů pro eliminaci tohoto rizika je hodně: použití datové struktury nenáročné na paměť, použití řídké matice, procedura zpracovávající pixel by měla po jeho zpracování okamžitě uvolnit paměť pokud už není potřeba atd.

2.2 Obecný postup

Existuje mnoho postupů při zvýraznění detailů v obraze. Každý postup se ovšem liší v použití odlišných druhů technik, které detekují detaily v obraze a/nebo je následně zvýrazňují. Techniky mohou také před samotným zpracováním eliminovat nežádoucí prvky, jakými může být mlha, rozostřený obrázek aj. Z těchto důvodů neexistuje (prozatím) žádný univerzální postup, který by dokonale zvýraznil detaily u jakéhokoli obrazu, protože techniky metod pro zvýraznění detailů v obraze se značně liší. Všeobecně ale platí následující postup:

- Předcházející úprava obrazu
- Zvýraznění detailů obrazu

2.2.1 Předcházející úprava obrazu

Předzpracováním obrazu se rozumí především odstranění nežádoucích prvků, neboli šumu. Šum znamená znečištění. U digitálního obrazu se může objevit jak jednobarevné (obrazy ve stupních šedi) nebo různobarevné zrnění. Šum se někdy nemusí podařit úplně potlačit a k jeho potlačení je potřeba především znalost původu signálu (v tomto případě obrazu) a charakteristiky šumu. K odstranění šumu v obraze může posloužit vhodné zvolení barevného modelu, ale nemusí to vždy stačit. Jedna ze složitějších metod určená k potlačení šumu je například 2D konvoluce pomocí **Gaussovského filtru**, určeného především k odstranění Gaussova šumu, kde je každý pixel v obraze mírně pozměněn (oproti náhodnému šumu, který je způsoben například špatnou technologií při pořízení snímku). Maska tohoto filtru je složena z elementů, které jsou dány Gaussovskou funkcí. Rovnice 2D Gaussovy funkce je zapsána následovně:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.1)$$

Metod existuje samozřejmě spousta dalších. Pro odstranění náhodného šumu se používá tzv. mediánový filtr, který pracuje na principu, že vezme pro každý pixel jeho okolí a

z okolí vybere medián, kterým nahradí hodnotu v pixelu. Existuje také více modifikací mediánového filtru, kdy se vezme například vážený průměr pixelů, nebo hodnota ležící ve dvou třetinách.



Obrázek 2.2: Příklady použití šumu a jejich potlačení. (nahore potlačení Gaussovkého šumu Gaussovým filtrem, dole potlačení "náhodného"šumu mediánovým filtrem)

2.2.2 Zvýraznění detailů v obraze

Po případném potlačení šumu v obraze nastává konečná fáze, kterou je zvýraznění detailů v obraze. Metody pro zvýraznění detailů mohou být jak jednokrokové, tak i vícekrokové.

2.3 Výhody a nevýhody

Jako cokoliv jiného, mají i metody zvýrazňující detaily v obraze své výhody i nevýhody. Jelikož výhody a nevýhody jedné metody se mohou lišit (ať už výrazně či málo) od ostatních, je těžké sepsat je sepsat dohromady. Mezi nejvýznamnější výhody týkající se obecného zvýraznění detailu v obraze se řadí tyto:

- Zvýšení množství informace v obraze
- Zlehčení dalšího zpracování obrazu (segmentace, klasifikace objektů)
- Zvýšení kontrastu v rozdílných oblastech obrazu

Na druhou stranu, nejčastějšími nevýhodami těchto metod jsou:

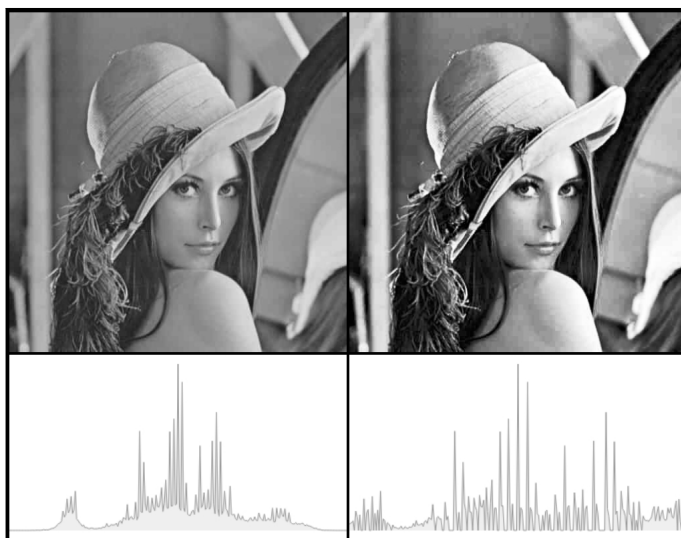
- Předzpracování obrazu
- Případná nutnost použití více odlišných technik
- Některé metody jsou časově a/nebo paměťově náročné na zpracování obrazu

2.4 Použití

Jestliže je množství informace v obraze po zvýraznění jeho detailů větší, než je v původním obraze, dokáže si člověk všimnout i nepatrných detailů, kterých by si v původním nezpracovaném obraze všimnout nemusel. Při zpracování se zvyšuje především rozdíl kontrastu mezi detailem a jeho okolím pro jeho zvýraznění. Zvýrazněný obraz může mít použití v mnohých oblastech. Mezi tyto oblasti můžeme počítat strojírenský průmysl, zdravotnictví, kriminalistika, architektura, umění a mnoho dalších. Příkladů konkrétního využití v jednom z oborů zmíněných v předešlé větě jsou spousty. Například ve strojírenském průmyslu lze lépe nalézt vadu na výrobku; v kriminalistice, konkrétně daktyloskopii, se více zvýrazní otisk prstu nebo v umění dojde k restauraci poškozeného obrazu nebo se pomocí těchto metod vytvoří umělecká fotografie, jejíž efekt by se nemusel zachytit na nezpracované fotografii pořízené běžným fotoaparátem.

2.5 Ukázka jednoduché metody - Ekvalizace histogramu

Histogram představuje rozložení četnosti výskytu barev v daném obraze. Z histogramu můžeme určit například prahovou hodnotu, která se používá při prahování popsaném v kapitole níže. Úprava (ekvalizace) histogramu je algoritmus, který upraví rozložení četnosti výskytu barev v obraze tak, aby se různé intenzity barev zobrazily v přibližně stejné četnosti a širším rozmezí. Díky tomu ekvalizace umožňuje zvýraznit v obraze špatně viditelné detaily.



Obrázek 2.3: Obrázek před a po ekvalizaci histogramu.

Kapitola 3

Popis vybraných metod - teoretická část

Tato kapitola se zabývá dvěma složitějšími metodami pro zvýraznění detailů v obraze z pohledu teoretické části. Především bude popsáno, na jakém principu jednotlivé metody fungují, jejich výhody a nevýhody a použití.

3.1 Art-Photographic Detail Enhancement

K vytvoření této metody se její tvůrci nechali inspirovat v oblasti uměleckých fotografií. Na rozdíl od jiných metod, u kterých je primární pouze změna tónu, metoda představuje nový model transformace, který se skládá ze dvou operátorů, kterými jsou posunutí a měřítko. Model umožňuje změnu tonálního rozsahu v každé oblasti obrazu, což umožňuje výrazné zvýšení detailu.



Obrázek 3.1: Ukázkový obrázek po použití metody. Zdroj obrázku [16]

3.1.1 Stručný popis

Hlavní cíl metody je dekompozice obrazu na základní a detailní vrstvu. Základní vrstva obsahuje velkokapacitní prvky a detailní vrstva obsahuje i prvky, kde obsahují i velmi jemné změny intenzity. Zvýraznění detailů je potom pouhá úprava detailní vrstvy před tím, než se propojí se základní vrstvou. Tímto způsobem se vytvoří obraz podobný umělecké fotografii, díky jehož zpracování se vyvolá estetický dojem z pohledu pozorovatele. Klasicky se tyto fotografie získávají způsobem, kdy fotograf nafotí více snímků za různých světelných podmínek a poté složí snímky dohromady v jeden celek. Další ze způsobů je pořízení snímků s vysokým dynamickým rozsahem (HDR) a poté vybrání nejvhodnější části snímků, které kvůli svým světelným podmínkám vyhovují pokaždé pro jinou oblast ve výsledném snímku. Tímto způsobem mohou být odhaleny i detaily ve světlých či tmavých oblastech. Klíčem k vytvoření těchto fotografií je tedy schopnost kontrolovat světelné podmínky pro každou oblast zvlášť.

Hlavní myšlenka pro zvýraznění detailů z jednoho LDR obrazu spočívá v sestavení modulu, který bude schopen transformovat jemné tóny v každém regionu v obraze, a to při zachování tónu v daném dynamickém rozsahu. Model transformace se skládá ze dvou operací, kterými jsou měřítko a posun. Jiné metody používají pouze jednu operaci v modelu transformace, kterou je pouze měřítko. Právě operace posunu umožňuje vytvoření základní vrstvy a hraje klíčovou roli při vytváření detailů ve velice tmavých či světlých oblastech. Pro výpočet těchto operací se používá optimalizační rámec s omezeními dynamického rozsahu.

3.1.2 Výhody a nevýhody

Hlavní výhodou této metody je především zvýraznění detailů ve tmavých a světlých oblastech. Jedna z dalších výhod je také skutečnost, že se zvýraznění detailů neaplikuje na obraz jako na celek, ale jako na zvýraznění detailů v jednotlivých oblastech, čímž se získá větší množství detailů. Značná nevýhoda této metody je enormní doba pro zpracování obrazu.

3.1.3 Použití

Jak už bylo zmíněno nahoře (3.1), obraz se zvýrazněnými detaily se může použít v oblasti umělecké fotografie, díky vytvoření "hyperrealistického" efektu fotografie. Z praktického hlediska se tato metoda může použít v lékařství při zvýraznění detailů v rentgenových snímcích, kde zvýraznění jejich detailů může urychlit či přesněji určit diagnózu pacienta. Kromě lékařství může být metoda použita také v určení rozsahu koroze na přístrojích, zjištění vad ve stavebních materiálech atd.

3.2 Contrast limited adaptive histogram equalization

Contrast limited adaptive histogram equalization (zkráceně CLAHE) vychází z obecné ekvalizace histogramu (2.5). Tato modifikace ekvalizace histogramu spočívá v rozdělení obrázku na jednotlivé oblasti (regiony) (4.2.3), upravení kontrastu každého regionu překlopením hodnot jeho histogramu (4.2.4) a ekvalizací histogramu z každého regionu tak, jak ji známe z obecné ekvalizace. Následující podkapitoly zahrnují stručný popis CLAHE, výhody a nevýhody vyplývající z této metody a v neposlední řadě i obrázky či oblasti, ve kterých lze tuto metodu použít.



Obrázek 3.2: Ukázka metody CLAHE. Zdroj obrázku [19]

3.2.1 Stručný popis

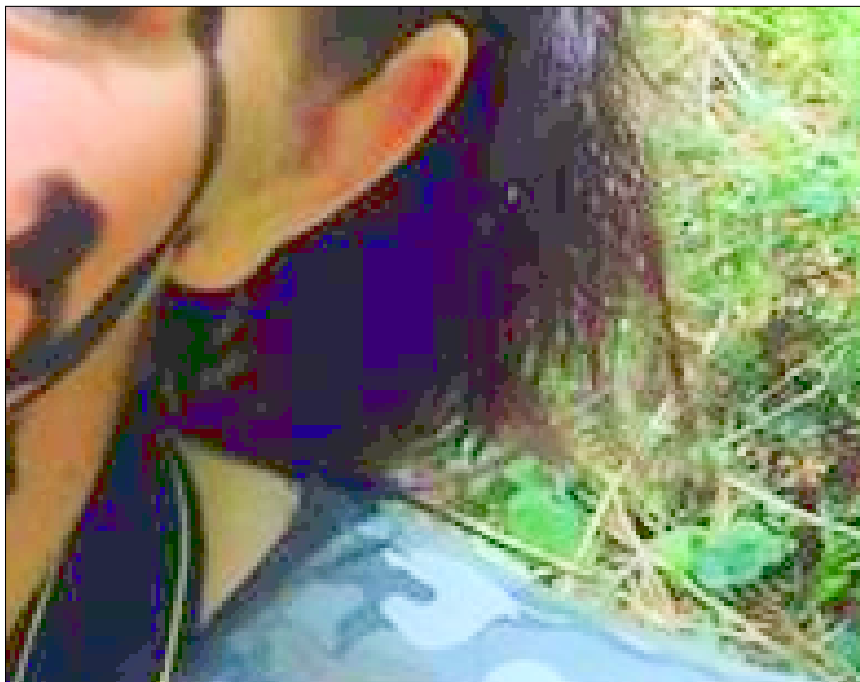
Jak už bylo zmíněno výše, původní obrázek vychází z obecnějších způsobů ekvalizace histogramu a zpracovává se pomocí několika procedur. Těmito procedurami jsou:

- Rozdělení obrazu na (přibližně) stejně velké oblasti (regiony)
- Překlopení histogramu vypočítaného v každém regionu pomocí tzv. "clip limitu"
- Vylepšená ekvalizace každého pixelu v regionu pomocí ekvalizovaných histogramů v regionu

Získání konečné hodnoty se počítá jinak, než v obecné ekvalizaci histogramu z důvodu, že je obraz ke zpracování rozdělený na jednotlivé regiony, a to podle toho, v jaké části obrazu se pixel nachází. Postup realizace zpracování pixelů je uveden v praktické části metod (4.2.3).

3.2.2 Výhody a nevýhody

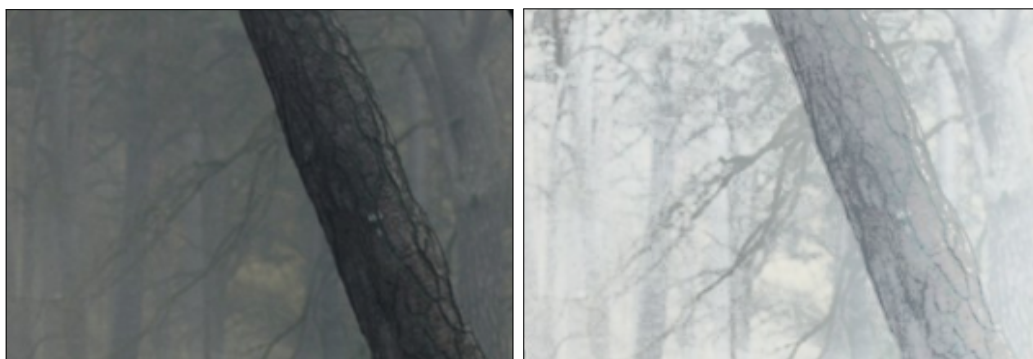
Jelikož se pomocí metody CLAHE nepoužívá ekvalizace histogramu celého obrazu, ale obraz je rozdělený na regiony, tak se obraz vylepší na lokální úrovni, a z toho důvodu můžou více vyniknout detaily ve velmi tmavých či velmi světlých oblastech. Metoda je také poměrně rychlá kvůli jednoduchým výpočtům pro získání správných hodnot pixelů. Značnou nevýhodou je to, že pro výsledný obraz se musí uložit do paměti histogram každého regionu, a to může být při malých rozměrech regionu na velkém obrázku velmi náročné na paměť. Proto musí být tyto histogramy uloženy v datovém typu, který je nenáročný na paměť. Další značnou nevýhodou je i to, že pokud je značná část regionu skládající se pouze z pixelů, které mají přibližně stejnou hodnotu barevné složky, tak se tyto barevné plochy kvůli ekvalizaci histogramu nezachovají a mohou vzniknout nežádoucí artefakty[13] (3.3), v tomto případě je nutné zaexperimentovat a postupně zvětšovat rozměry regionu, aby se tyto artefakty co nejvíce potlačily.



Obrázek 3.3: Detail nežádoucího artefaktu obrázku po ekvalizaci histogramu, pokud má region v místě artefaktu převážně stejnou hodnotu pixelu. Výřez z vlastního obrázku.

3.2.3 Použití

Jelikož se pomocí metody dají zvýraznit i detaily také ve velmi malých, či naopak ve velmi světlých oblastech, může se metoda použít ke zpracování špatně vyfoceného obrazu, který má spousty těchto tmavých či světlých oblastí. Může se jednat například o obrázky vyfocené na tmavých místech jako jsou lesy, jeskyně, místa v mlžném oparu a podobně. Do jisté míry lze metoda použít i pro zvýraznění detailů na rentgenových snímcích, ovšem za rizika zkreslení a tím i podávání nesprávné informace, která by mohla vést ke špatné diagnóze se tato metoda moc nepoužívá, případně se používají její další modifikace, které se snaží riziko zkreslení obrazu eliminovat.



Obrázek 3.4: Ukázka ekvalizace obrázku s mlžným oparem v lese. Původní obrázek převzat z [5]

Kapitola 4

Popis vybraných metod - praktická část

V této kapitole budou metody popsány z praktické části, bude podrobněji vysvětlen princip metod, a v neposlední řadě budou popsány vzorce či algoritmy těchto metod. Následující kapitoly jsou volně přeloženy, a tudíž nepřímou citují s drobnými úpravami článek Art-Photographic Detail Enhancement včetně jeho rovnic [15].

4.1 Art-Photographic Detail Enhancement

Praktický popis této metody je rozdělený do následujících tří podkapitol:

- Model tónové transformace
- Dekompozice základní a detailní vrstvy
- Maximalizace detailů

V podkapitolách jsou uvedeny vzorce pro implementaci této metody a jejich popis a důvod použití. V podkapitolách může být uveden také stručný popis technologie, která byla převzata z jiných publikací. Stručný popis těchto technologií je z důvodu, že tyto metody jsou rozsáhlé a že nespádají přímo pod oblast metod pro zvýraznění detailů.

4.1.1 Model tónové transformace

Originální obraz I se na začátku rozloží na základní vrstvu B (base layer), která je rozmazanou verzí originálního obrazu a detailní vrstvu D (detail layer), která se získá z rozdílu originálního obrazu I a základní vrstvy B . Jednoduchou cestou ke zvýraznění detailů by měly být koeficienty detailů D_i , které jsou v rovnici zobrazeny následujícím způsobem:

$$I'_i = B_i + D'_i = B_i + s_i D_i \quad (4.1)$$

kde s je faktor měřítka. Je však nutno dodat, že možný rozsah s_i je ohraničen hodnotou I_i , která nemůže překročit maximální dynamický rozsah na zobrazovacím zařízení. Tím by mohlo dojít k omezení zvýraznění detailů v tmavých a světlých oblastech. Proto se model transformuje do následující podoby:

$$I'_i = B_i + D'_i = (B_i + t_i) + s_i D_i \quad (4.2)$$

kde t_i určuje množství posunu aplikovaného na základní vrstvu B. Díky tomuto zásahu by se měly při pozitivním t_i rozjasnit tmavé oblasti v základní vrstvě a naopak, při negativním t_i se světlé oblasti v základní vrstvě ztmaví.

Tento model v podstatě ztělesňuje základní princip umělecké fotografie, který spočívá v nezávislé kontrole jasu v každé oblasti.

Pokud by byly t_i a s_i fixní pro všechny pixely v obraze I, tak by se zachovala většina z původní struktury scény, ovšem s velmi omezeným rozsahem pro zvýraznění detailů.

Proto se musí zajistit plynulá změna t_i a s_i s různými hodnotami měřítka pro každou oblast. Díky tomu, že má každá oblast jiný dynamický rozsah, je možné více zvýraznit detaily po celém obraze.

4.1.2 Dekompozice základní a detailní vrstvy

Ideální základní vrstva B získaná z I by měla mít konstantní tón, který je uvnitř každé homogenní oblasti obrazu a zároveň by měla zachovat tvar okrajů tak, aby se podobaly co nejvíce okrajům z originálního obrazu I. Proto existují dva důležité požadavky na základní vrstvu:

- nutný konstantní základ za pomoci stanovení hranic mezi jednotlivými heterogenními oblastmi
- odpovídající okraje k I pro předejití vzniku artefaktů při zvýraznění detailů

Získání základní vrstvy B z obrazu I spočívá ve třech krocích:

1. Vyhlazení obrazu za pomoci metody L0 smoothing
2. Vyhlazení obrazu za pomoci adaptivního L0 smoothing
3. Vyhlazení obrazu za pomoci adaptivního Gaussova rozmazání

L0 smoothing

Tato metoda generuje sadu konstantních oblastí a krokových hran, čímž splňuje první požadavek. Původní L0 smoothing řeší následující problém:

$$\min_B \left\{ \sum_i (B_i - I_i)^2 + \lambda C(B) \right\} \quad (4.3)$$

kde $C(B) = \#\{i \mid |\delta_x B_i| + |\delta_y B_i| \neq 0\}$ je funkce počítající nenulové gradienty. Tato minimalizace je řešená skrz speciální alternativní optimalizaci s pomocnými proměnnými h_i a t_i , které korespondují k $|\delta_x B_i|$ a $|\delta_y B_i|$, přesněji řečeno pomocí těchto dvou rovnic:

$$\min_B \left\{ \sum_i (B_i - I_i)^2 + \beta((\delta_x B_i - h_i)^2 + (\delta_y B_i - v_i)^2) \right\} \quad (4.4)$$

a

$$\sum_i \min_{h_i, v_i} \left\{ (h_i - \delta_x B_i)^2 + (v_i - \delta_y B_i)^2 + \frac{\lambda}{\beta} H(|h_i| + |v_i|) \right\} \quad (4.5)$$

kde $H(\|h_i\| + \|v_i\|)$ je binární funkce mající hodnotu 1, pokud je $\|h_i\| + \|v_i\| \neq 0$, v opačném případě má funkce hodnotu 0. λ funguje jako vyhlazovací parametr, který kontroluje počet

nenulových gradientů. Její hodnota byla nastavena jako v původní publikaci na 0.01. Pomocí této metody získáme vrstvu B^1 . Účelem aplikace L0 smoothingu je získání adaptivní λ mapy ze základní vrstvy B^1 vypočítané z prvního průchodu, která se použije při druhém průchodu. K implementaci této fáze byla použita knihovna z githubu[1], která už L0 Smoothing realizuje. Některé funkce této knihovny byly pro účely implementace (převážně pro realizaci druhé fáze) metody upraveny [18].

Adaptivní L0 smoothing

V tomto průchodu použijeme upravenou metodu L0 smoothing, která používá pro každý pixel v obraze jinou hodnotu λ získanou z adaptivní λ mapy. Tato mapa se získá následujícím způsobem. Nejprve je vypočítán gradient G^1 z B^1 . K vypočítání gradientu byla použita funkce z původní L0 Smoothing knihovny. Poté, co jsme získali gradient, tak pomocí něj získáme λ pro každý pixel v obraze tak, že použijeme následující funkci:

$$\rho(u, \sigma) = \begin{cases} \frac{1}{3} & \text{if } u < -\sigma \\ \frac{u^2}{\sigma^2} - \frac{u^4}{\sigma^4} + \frac{u^6}{3\sigma^6} & \text{if } -\sigma \leq u < 0 \\ 0 & \text{if } u \geq 0 \end{cases} \quad (4.6)$$

Po získání $\rho(u, \sigma)$ dostaneme λ pro každý pixel v obraze následující rovnicí:

$$\lambda_i = 3(\lambda - \eta)\rho(G_i^1 - \alpha, \sigma) + \eta \quad (4.7)$$

kde G_i^1 značí gradient pixelu. Hodnota σ je nastavena na 0.1, hodnota α je nastavena na 0.2 a hodnota η je nastavena na číslo, které by mělo být co nejblíže nule. Poté, co získáme mapu, provedeme na původní obrázek I adaptivní L0 Smoothing, který tuto mapu využívá. Funkce pro adaptivní L0 Smoothing je vlastně upravená původní funkce pro L0 Smoothing obohacená o parametr *adaptiveLambdaMap*, který označuje mapu λ hodnot pro každý pixel. Tím získáme vrstvu B^2

Adaptivní Gaussovo rozmazání

K adaptivnímu Gaussovu rozmazání použijeme originální obraz I a vrstvu B^2 získanou z předchozího průchodu. Toto rozmazání spočívá ve filtraci obrazu, která nemá stejný filtr pro všechny pixely, ale má pro každý pixel odlišný filtr. Pro velikost každého gaussovského filtru je potřeba znát jeho hodnotu σ , která je jiná pro každý pixel. Tyto parametry získáme pomocí optimalizace následující rovnice:

$$\min_{\sigma} \left\{ \sum_i ((G(\sigma_i) \star S) - I_i)^2 + \gamma((\delta_x \sigma_i)^2 + (\delta_y \sigma_i)^2) \right\} \quad (4.8)$$

kde $G(\rho_p)$ je nenulový Gaussian se standardní odchylkou σ_p . U barevných obrazů je pro správnou minimalizaci funkce potřeba součtu všech barevných kanálů. První část rovnice $((G(\sigma_i) \star S) - I_i)^2$ vynucuje podobnost mezi rozmazaným a originálním obrazem. Druhá část rovnice $(\delta_x \sigma_i)^2 + (\delta_y \sigma_i)^2$ omezuje σ mapu kvůli zabránění ojedinělým šumům. Hodnota η je nastavena na 0.001.

Abychom mohli prakticky spočítat tuto rovnici, přiřadíme k σ_i diskrétní hodnoty (0, 1/3, ..., 3). Poté nastavíme velikost filtru na $6\sigma + 1$. Tímto způsobem můžeme řešit optimalizační funkci klasickými metodami. Po získání parametrů této mapy použijeme konvoluci na originální obraz za účelem získání konečné vrstvy B .

4.1.3 Maximalizace detailů

Po získání jak základní, tak detailní vrstvy z obrazu je potřeba optimalizovat parametry s_i a t_i tak, aby bylo zvýrazněno tolik detailů, kolik je možných. Pro získání těchto parametrů musíme použít optimalizaci kvadratické funkce.

Získání vah

Aby byly zavedena kontrola detailů po částech, musí se použít váhy, které vycházejí z gradientů získané základní vrstvy. Hodnota vah se pohybuje v rozsahu $w_i \in [0, 1]$. Dále si definujeme $w_i = K(\nabla B_i)$, kde $K(u)$ je Tricubická funkce. Hodnoty jednotlivých vah se vypočítají následující rovnicí:

$$w_i = K(\nabla B_i) = \begin{cases} (1 - |\frac{\nabla B_i}{a}|^3)^3 & \text{if } \nabla B_i \leq 1 \\ 0 & \text{jinak} \end{cases} \quad (4.9)$$

Parametr a je nastaven na hodnotu 0.2. Pomocí vah dokáže objektivní funkce poznat, že se pixel nachází v homogenním regionu.

Objektivní funkce

Objektivní funkce k maximalizaci detailů je zapsána tímto způsobem:

$$\min_{s,t} \left\{ - \sum_i \|s_i D_i\|^2 + r_1 \sum_i w_i \|\nabla s_i\|^2 + r_2 \sum_i w_i \|\nabla t_i\|^2 \right\} \quad (4.10)$$

s podmínkou:

$$0 \leq I'_i = (B_i + t_i) + s_i D_i \leq 1 \quad (4.11)$$

Tato podmínka zamezí, aby se hodnoty s_i a t_i neblížily do nekonečna a je díky ní zpracovaný pixel v daném barevném rozsahu. Hodnoty r_1 a r_2 jsou nastaveny na 200 a 500. Aby jsme omezili konvexní problémovou podmínku, bez které by kvadratické programování nedokázalo v objektivní funkci nalézt minimum, omezíme $r * w_i$ tak, aby bylo větší než 2, na $w_i = K(\nabla B_i) + 2/r$. Optimalizace této rovnice je realizována pomocí knihovny CGAL (B.2), která je rychlá, přesná a podporuje kvadratické programování.

Implementace objektivní funkce do CGALu

K tomu, aby optimalizace této objektivní funkce byla realizována pomocí knihovny CGAL, musí se převést do obecné formy, která řeší následující problém:[3]

$$\begin{aligned} \min_x \quad & x^T D x + c^T x + c_0 \\ \text{s.t.} \quad & A x \leq b \\ & l \leq x \leq u \end{aligned}$$

v n reálných proměnných $\mathbf{x} = (x_0, \dots, x_{n-1})$, kde

- A je $m \times n$ matice obsahující podmínky,
- b je m -dimenzionální vektor pravé strany
- \leq je m -dimenzionální vektor vztahů

- l je n -dimenzionální vektor spodních hranic x , kde $l_j \in \mathbb{R} \cup \{-\infty\}$ je platný pro všechny j
- u je n -dimenzionální vektor horních hranic x , kde $u_j \in \mathbb{R} \cup \{\infty\}$ je platný pro všechny j
- D je symetrická pozitivně semidefinitní $n \times n$ matice pro kvadratické programování
- c je n -dimenzionální vektor pro lineární programování
- c_0 je konstanta

Nutno podotknout, že pokud D je rovno nule, tak se jedná o lineární program, tudíž se CGAL může použít i pro řešení lineární rovnice. Převod objektivní funkce (4.10) do maticového tvaru je popsán na následujících řádcích. Mějme následující matici, která představuje obrázek, a jednotlivé prvky této matice znázorňující pixely.

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

Tabulka 4.1: Tabulka znázorňující vzorový obrázek

Jelikož vložení parametrů s a t do matice je velice podobné, bude zde podrobněji popsána pouze implementace prvního druhu parametrů. Převod první části rovnice (tj. $-\sum_i \|s_i D_i\|^2$) do maticového tvaru je jednoduché, protože v této části parametry s nejsou na sebe navzájem závislé. Problém nastává u převodu druhé části rovnice (tj. $\sum_i w_i \|\nabla s_i\|^2$) do maticového tvaru, protože parametry na sebe mohou navzájem navazovat. Konkrétně jde o $\|\nabla s_i\|^2$, které vyjadřuje gradient daného parametru rozepsaný do rovnice:

$$\|\nabla s_i\|^2 = (s_{i-1} - s_{i+1})^2 + (s_{i-width} - s_{i+width})^2 \quad (4.12)$$

, kde $width$ značí šířku obrazu. Pro převod této části rovnice do maticového tvaru je potřeba pro každý parametr s vědět, ve kterých pixelech se používá a také s jakými jinými parametry se násobí. Pro výpočet gradientů pixelů, které jsou v rozích a na krajích obrazu je nutné upravit matici zkopírováním jejich hodnoty do "pomocných buněk". Tyto buňky jsou zvýrazněny v této tabulce:

-	A	B	C	D	E	-
A	A	B	C	D	E	E
F	F	G	H	I	J	J
K	K	L	M	N	O	O
P	P	Q	R	S	T	T
U	U	V	W	X	Y	Y
-	U	V	W	X	Y	-

Tabulka 4.2: Tabulka znázorňující vzorový obrázek se zkopírovanými hodnotami (červené písmo) pro nalezení gradientů ve všech pixelech

Po této úpravě tabulky dokážeme spočítat i gradienty, které se vyskytují v rozích či po stranách. Pro příklad, $\|\nabla A\|^2$ je vypočítán jako $(A - B)^2 + (A - F)^2$ pro pixel vyskytující se v rohu obrazu, $\|\nabla K\|^2$ je vypočítán jako $(K - L)^2 + (F - P)^2$ pro pixel vyskytující se na kraji obrazu, a pro ostatní pixely, např. $\|\nabla M\|^2$ je vypočítán jako $(L - N)^2 + (H - R)^2$. Tímto jednoduchým způsobem jsou zajištěny všechny gradienty pixelů pro vepsání do matice pro optimalizaci.

Jelikož pixely v rozích nebo po stranách mají trochu odlišné pravidla pro vytvoření gradientu a také, protože program zpracovává obraz po jednotlivých pixelech, je nutné stanovit obecná pravidla pro nalezení všech částí, ve kterých se parametr s_i vyskytuje (tj. všech částí pixelů obrazu, se kterými parametr, který chceme vepsat do matice komunikuje) a poté jednotlivé složky matice D správně sečíst, aby byla hodnota parametrů vypočítána správně.

Při experimentování se vzorovým obrazem (tj. nalezení těch pixelů, ve kterých se parametr s_i reprezentující daný pixel po rozepsání gradientu daného pixelu vyskytuje nebo ne), byla rovnice pro vytvoření matice vedoucí ke správné optimalizaci objektivní funkce zobecněna do následující tabulky podmínek, ve kterých se parametr s_i vyskytuje (tj. se kterými dalšími parametry komunikuje):

Podmínka	Přičti k D-čkové složce s_i	Přiřaď k s_i tento prvek pro násobení
$i > 1$	$2 * r_1 * w_{i-1}$	s_{i-2}
$j < height - 2$	$2 * r_1 * w_{i+width}$	$s_{i+2*width}$
$i < width - 2$	$2 * r_1 * w_{i+1}$	s_{i+2}
$j > 1$	$2 * r_1 * w_{i-width}$	$s_{i-2*width}$
$i == 0$	$2 * r_1 * w_i$	s_{i+1}
$i == 1$	$2 * r_1 * w_{i-1}$	s_{i-1}
$i == width - 1$	$2 * r_1 * w_i$	s_{i-1}
$i == width - 2$	$2 * r_1 * w_{i+1}$	s_{i+1}
$j == 0$	$2 * r_1 * w_i$	$s_{i+width}$
$j == 1$	$2 * r_1 * w_{i-width}$	$s_{i-width}$
$j == height - 1$	$2 * r_1 * w_i$	$s_{i-width}$
$j == height - 2$	$2 * r_1 * w_{i+width}$	$s_{i+width}$

Tabulka 4.3: Tabulka znázorňující podmínky pro sestavení matice

Díky této tabulce je jasné, se kterými váhami daný parametr komunikuje, což je potřeba pro vyhotovení matice D a je také možné zavést správné součty násobků parametrů, které mají být umocněné do matice D . Násobky pro umocnění matice získané z tabulky také značí hodnotu, kterou jsou vynásobeny prvky matice, pokud jsou násobeny mezi sebou (viz. druhý sloupec tabulky). Matice ovšem musí být pozitivně semidefinitní, proto je potřeba pro jistotu matici D vynásobit s její transponovanou formou. Po této operaci je zajištěno, že matice D je pozitivně semidefinitní [7]. Je to z toho důvodu, že pokud matice D je pozitivně semidefinitní, pak se jedná o úlohu konvexního kvadratického programování a řešení problému je podobné lineárnímu programování [12] a je řešen právě konvexní problém (4.1.3). Po vyhotovení matic podle tabulky výše zbývá nastavit podmínky, které zajistí, aby vypočítané hodnoty parametrů aplikované na původní základní a detailní vrstvu za účelem získání výsledného obrázku byly v daném barevném rozsahu. K tomu je potřeba převést podmínku (4.11)

$$0 \leq (B_i + t_i) + s_i D_i \leq 1 \quad (4.13)$$

do tvaru, který jde vložit do matice A a konstanty b . K tomu, aby byla podmínka pro vložení do výše zmíněné matice a konstanty uskutečnitelná, je potřeba výše zmíněnou podmínku rozdělit a upravit. Po rozdělení a úpravě nám vzniknou dvě samostatné podmínky ve tvaru

$$t_i + s_i D_i \leq 1 - B_i \quad (4.14)$$

a

$$-t_i - s_i D_i \leq B_i \quad (4.15)$$

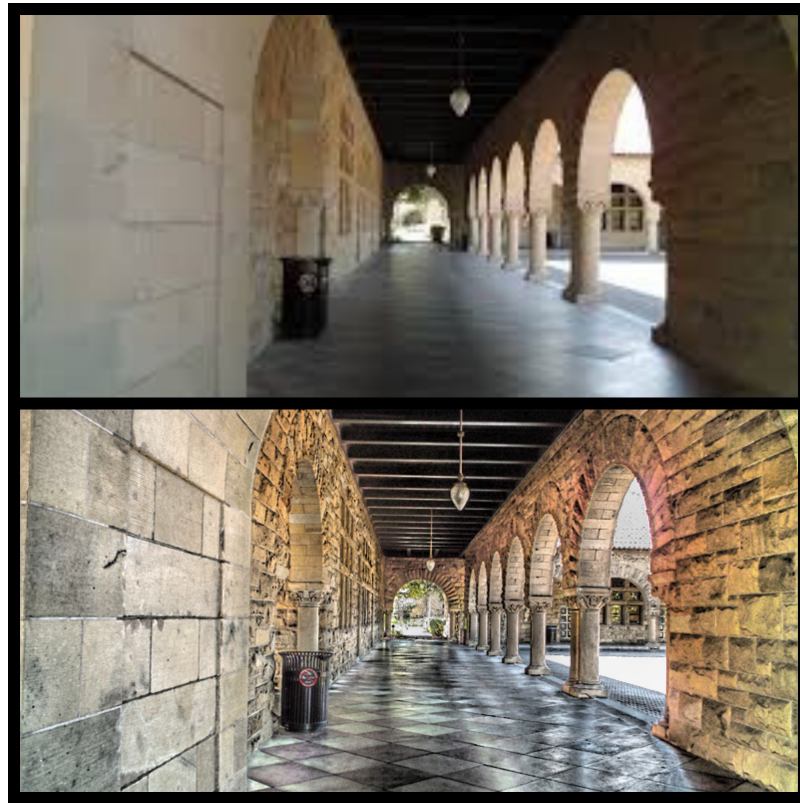
. Po této aritmetické úpravě rovnic je už jednoduché podmínky převést do tvaru, který podporuje knihovna CGAL (tj. $Ax \leq b$). Je nutné také podotknout, že celkově bude pro každou dvojici parametrů celkem šest podmínek (3 pro rovnici (4.14) a 3 pro rovnici (4.15)), protože se zpracovává obraz v barevném modelu RGB, který má tři barevné kanály.

Ovládání míry množství detailů

K tomu aby byly detaily vloženy v míře, jaké uživatel chce, je potřeba následující rovnice:

$$I_\mu = (\mu s + (1 - \mu))D + B + \mu t \quad (4.16)$$

Hodnota μ se pohybuje v rozmezí 0 a 1. Právě pomocí této hodnoty kontrolujeme množství detailů v obraze.



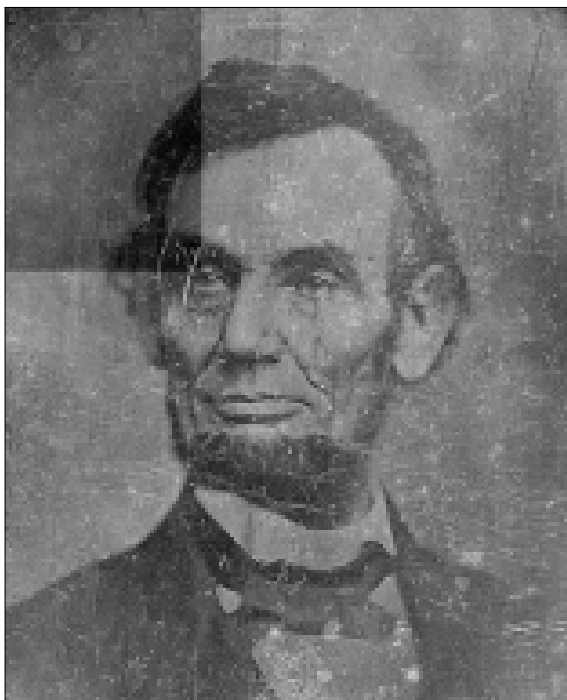
Obrázek 4.1: Původní obrázek (nahore) a obrázek po maximalizaci detailů (dole). Obrázek převzat z [16]

4.2 Contrast limited adaptive histogram equalization

V této části jsou popsány jednotlivé kroky, pomocí nichž je CLAHE realizován. V podkapitolách jsou kroky detailně popsány, uvedeny postupy pro jejich realizaci a v neposlední řadě také příkladové obrázky po zpracování jednotlivými kroky. Jako ukázkový příklad byl použit barevný obrázek a obrázek ve stupních šedi.



Obrázek 4.2: Vstupní barevný obrázek. Zdroj obrázku nenalezen.



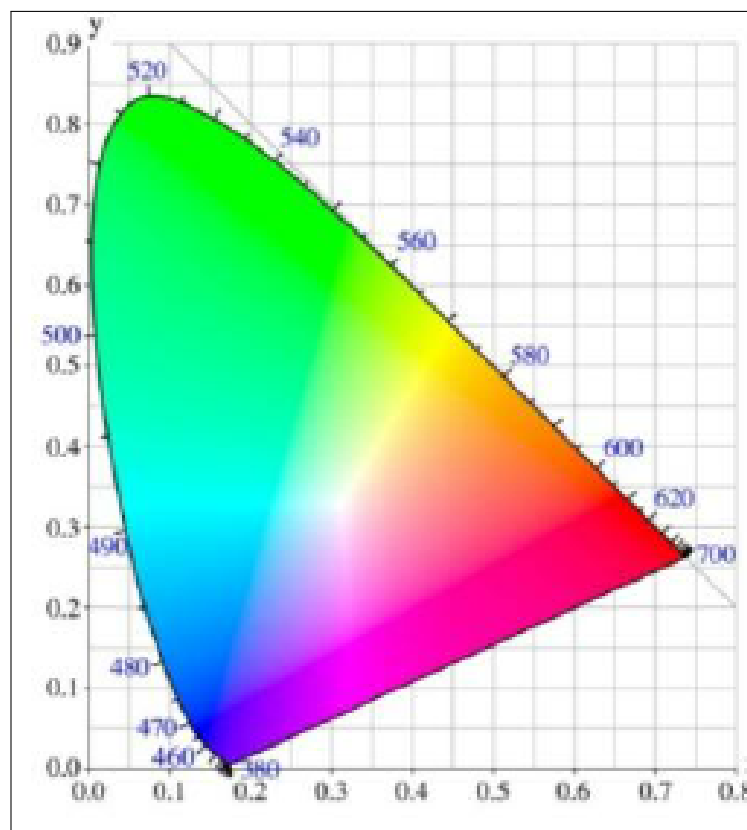
Obrázek 4.3: Vstupní obrázek ve stupních šedi. Zdroj obrázku [10]

4.2.1 Yxy barevný model

Tento barevný model má jasovou složku Y a barvy jsou určeny pomocí parametrů x a y . Díky oddělení jasové složky od barevných jsou kanály na sobě nezávislé. Proto se model

používá především ve zpracování digitálního obrazu (oproti modelu RGB, který slouží k zobrazení obrazu).

Yxy barevný model byl použitý z důvodu, že ekvalizace histogramu zahrnuje pouze ekvalizaci hodnot jasové složky. Pokud by se ekvalizace histogramu aplikovala jednotlivě pro každý kanál modelu RGB, tak by mohlo dojít k narušení barevné rovnováhy obrazu.



Obrázek 4.4: Ukázka barevného modelu Yxy. Zdroj obrázku [9]

4.2.2 Histogram equalization

Stručně popsaná ekvalizace histogramu byla popsána zde (2.5). V této kapitole bude ekvalizace histogramu popsána podrobněji. Obecná ekvalizace histogramu spočívá ve třech krocích:

1. Vytvoření histogramu
2. Vytvoření kumulativní distribuční funkce (histogramu)
3. Transformace kumulativní distribuční funkce

Vytvoření histogramu

Vytvoření histogramu spočívá v rozdělení stejných hodnot obrazu do odpovídajících tříd (anglicky bin). Jedná se o jednoduchý součet stejných hodnot pro každou třídu (2.5). Pro ověření správnosti implementace histogramu by měl součet četností odpovídat počtu pixelů.

Toto platí u diskrétních hodnot (RGB). Pro vytvoření histogramu zobrazující četnosti spojitých hodnot se nepočítá o součet stejných hodnot, protože by diagram mohl mít enormní množství tříd. Z tohoto důvodu nejsou jednotlivé třídy zastoupeny jako hodnoty pixelů, ale jako intervaly o stejném rozsahu. Celkový součet intervalu tříd udává celkový rozsah hodnot, konkrétně jde o rozsah od minimální po maximální hodnotu pixelu. Intervaly jednotlivých tříd jsou poté vypočítány jako podíl rozdílů maximální a minimální hodnoty a počtu tříd (v tomto případě 256). Následující algoritmus nalezne maximální a minimální hodnotu a poté vytvoří histogram ze spojitých hodnot.

Data: *inputImage*
Result: *outputHist*

```
//nalezení maximální, respektive minimální hodnoty v obrázku
minValue ← E90;
maxValue ← 0 ;
for j = 0; j < inputImage.rows; j++ do
    for i = 0; i < inputImage.cols; i++ do
        if inputImage(j,i) < minValue then
            | minValue ← inputImage(j,i)
        end
        if inputImage(j,i) > maxValue then
            | maxValue ← inputImage(j,i)
        end
    end
end
subValue ← (maxValue - minValue)/256
//v případě spojitých hodnot, je potřeba ošetřit maximální hodnotu obrázku tímto
způsobem
for j = 0; j < inputImage.rows; j++ do
    for i = 0; i < inputImage.cols; i++ do
        if inputImage(j,i) == maxValue then
            | outputHist[255]++;
        end
        else
            | outputHist[(int)(inputImage(j, i)/subValue)]++;
        end
    end
end
end
```

Algoritmus 1: Vytvoření histogramu ze spojitých hodnot, obdoba vytvoření histogramu z diskrétních hodnot (4.2.2)

Vytvoření kumulativní distribuční funkce

Kumulativní distribuční funkce vyjadřuje pravděpodobnost výskytu hodnoty obsažené v intervalu vzhledem k histogramu a je nezbytná pro poslední krok, aby byl získán správně

vyrovnaný histogram. Vychází z této rovnice[17]:

$$c(I) = \frac{1}{N} \sum_{i=0}^I h(i) = c(I-1) + \frac{1}{N} h(I) \quad (4.17)$$

kde N je počet pixelů v obraze, $h(i)$ je četnost i -té třídy histogramu a I je počet tříd daného histogramu.

Transformace kumulativní distribuční funkce

Pro získání ekvalizovaného histogramu je potřeba transformovat kumulativní distribuční funkci z předešlého kroku tak, že každý prvek z distribuční funkce se vynásobí rozdílem obsahu hodnot a ke kterému je posléze přičtena minimální hodnota. Rovnice hodnoty i -tého prvku nového histogramu je vypočítána následovně:

$$h'(i) = c(i) * (max - min) + min \quad (4.18)$$

Po ekvalizaci histogramu se obrázek zpracuje podle nového histogramu pomocí následujícího algoritmu:

Data: *transformedHist, inputImage, maxValue, subValue*

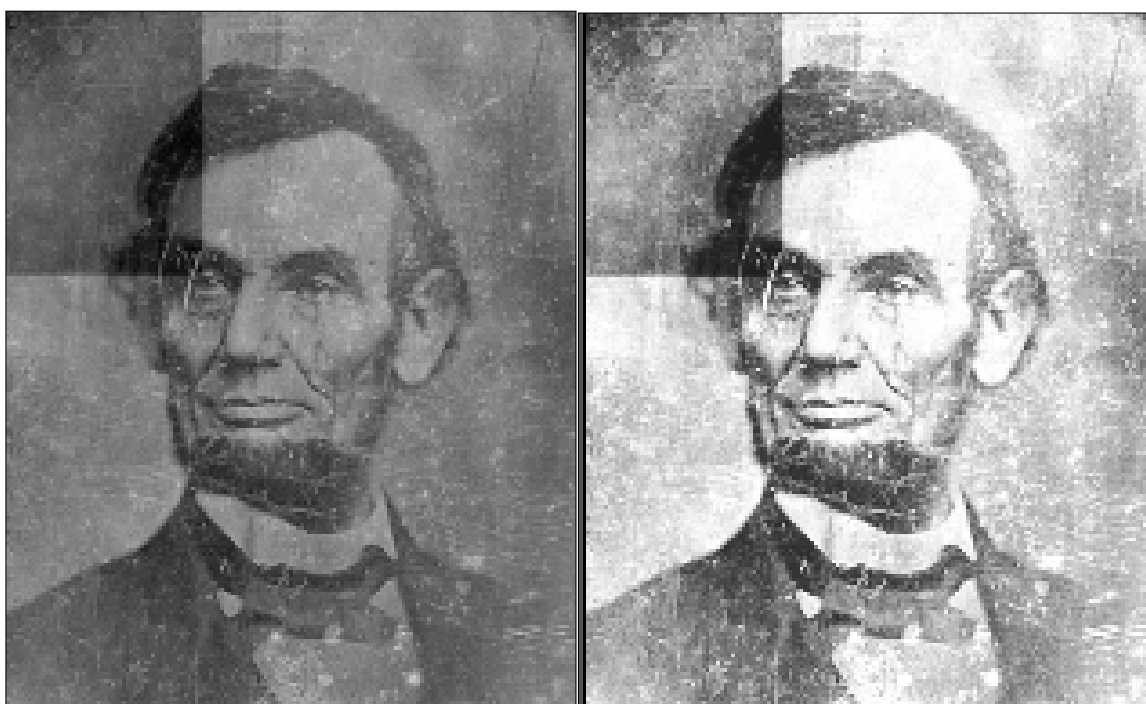
Result: *newImage*

```
for j = 0; j < inputImage.rows; j++ do
    for i = 0; i < inputImage.cols; i++ do
        //v případě spojitých hodnot, je potřeba ošetřit maximální hodnotu obrázku
        tímto způsobem
        if inputImage(j,i) == maxValue then
            | newImage(j,i) = transformedHist[255];
        end
        else
            | newImage(j,i) = transformedHist[(int)(floor)(inputImage(j,
            | i)/subValue)]
        end
    end
end
end
```

Algoritmus 2: Získání histogramu ze spojitých hodnot



Obrázek 4.5: Vstupní barevný obrázek (vlevo) a výsledný obrázek po ekvalizaci



Obrázek 4.6: Vstupní obrázek v úrovních šedi (vlevo) a výsledný obrázek po ekvalizaci

Výše jsou ukázány příklady ekvalizace obrázku ve stupních šedi a barevného obrázku. Obrázek ovšem může obsahovat oblasti, které jsou výrazně světlejší nebo tmavší, než většina snímků. Z tohoto důvodu nemusí být v těchto oblastech kontrast dostatečně vylepšen, a proto se používá modifikace metody, známé jako Adaptive histogram equalization, k odstranění tohoto problému.

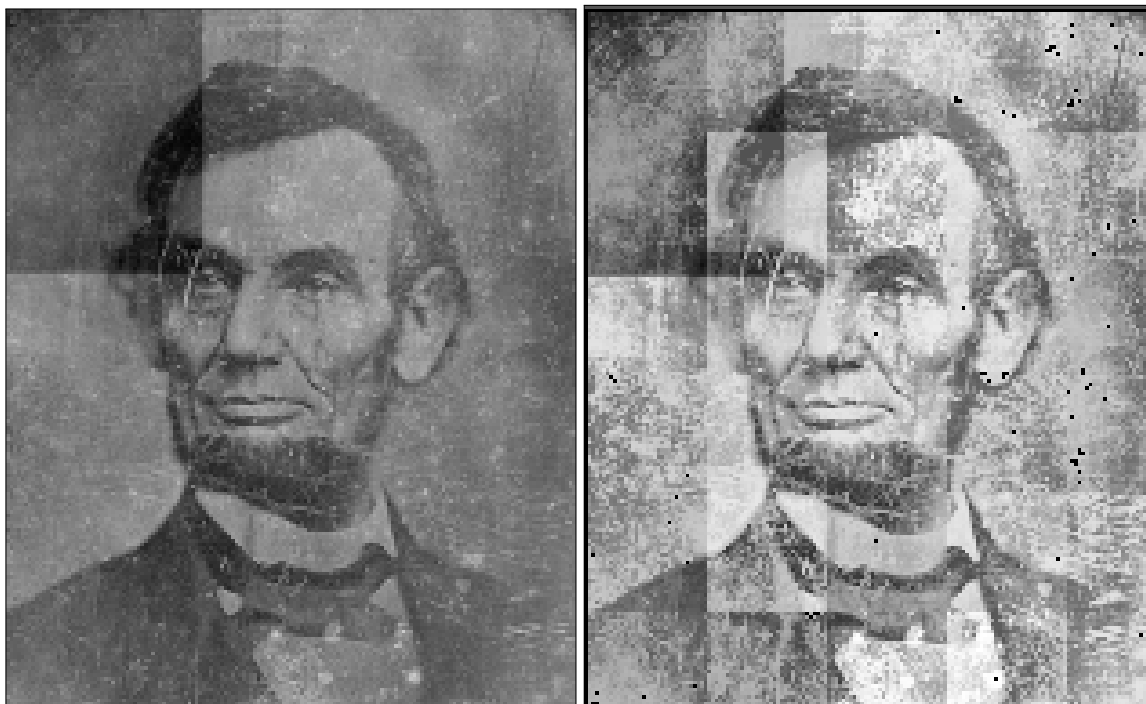
4.2.3 Adaptive histogram equalization

Myšlenka adaptivní ekvalizace histogramu spočívala v rozdělení obrázku na regiony a ty posléze zpracovat pomocí klasické ekvalizace. Tato technika poté odstraní nežádoucí artefakty v podobě nedostatečného vylepšení kontrastu. Samotná ekvalizace jednotlivých regionů ovšem nestačí, protože je vysoká pravděpodobnost vzniku viditelných hranic jednotlivých

regionů. Proto je nutné použít pro získání plynulého přechodu mezi jednotlivými regiony bilineární a lineární interpolaci.



Obrázek 4.7: Původní barevný obrázek a transformovaný obrázek bez interpolací



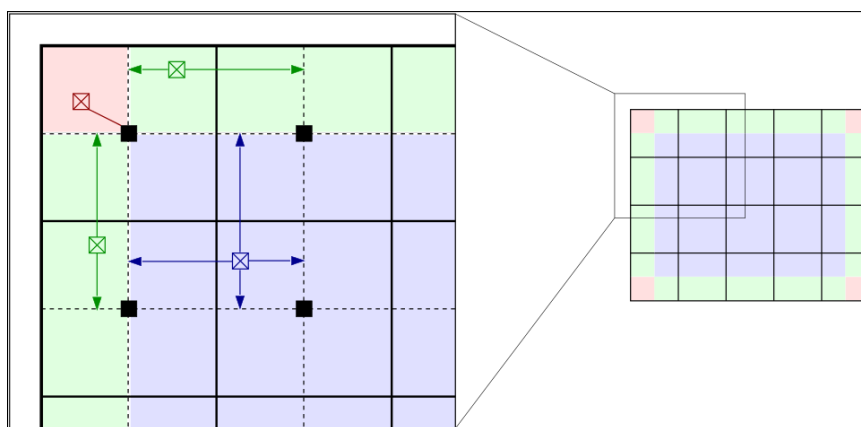
Obrázek 4.8: Původní obrázek ve stupních šedi a transformovaný obrázek bez interpolací

Mapa regionů

Obrázek (4.9) znázorňuje obrázek určený k ekvalizaci, který slouží jako ilustrace k tomu, aby bylo zřetelněji vidět, jakým způsobem se zpracovávají jednotlivé pixely. Bilineární či lineární interpolace se používá pouze v určitých částech obrázku a proto je nutné zjistit, u jakých pixelů se tyto procedury použijí. Vstupní obrázek ke zpracování je rozdělen na regiony, které jsou poté rozděleny "významnými pixely", tj. pixely, které se nacházejí ve střezech regionů (černý čtvereček). Podle polohy pixelu se určí, kterou ze tří procedur se pixel zpracuje. Těmito procedurami jsou:

- Transformační funkce
- Bilineární interpolace
- Lineární interpolace

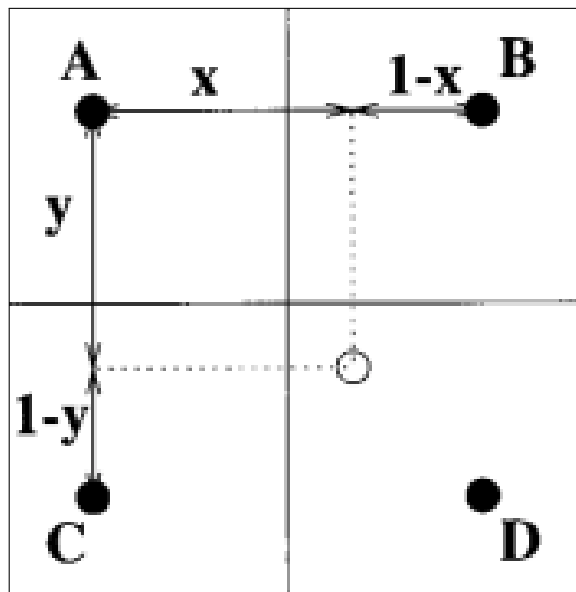
Pokud je pixel v růžové oblasti, ke zpracování pixelu se použije transformační funkce (4.2.2). V případě, že pixel leží v jedné ze zelených oblastí, se použije lineární interpolace (pixel leží mezi dvěma významnými body) (4.2.3). V posledním případě, kdy je pixel v modré oblasti, se k jeho zpracování použije bilineární interpolace (4.2.3). V následujících odstavcích jsou poslední dvě procedury zmíněné výše popsány detailně.



Obrázek 4.9: Mapa regionů obrázku a jeho významné body. Zdroj obrázku [2]

Bilineární interpolace

Bilineární interpolace se používá v nalezení hodnoty bodu (dále pixelu), který se nachází v pravidelné prostorové mřížce (tj. pravidelný čtyřúhelník), jehož hodnoty vrcholů jsou známy.



Obrázek 4.10: Bilineární interpolace. Zdroj obrázku [8]

Hodnota pixelu (průhledný puntík, dále označený jako s'), který leží mezi známými body (A, B, C a D) se určí následující rovnicí:

$$s' = (1 - y)((1 - x)g_A(s) + xg_B(s)) + y((1 - x)g_C(s) + xg_D(s)) \quad (4.19)$$

kde s je původní hodnota pixelu, s' je nová hodnota pixelu a $g_A(s)$, $g_B(s)$, $g_C(s)$ a $g_D(s)$ jsou mezihodnoty pixelu s , vypočítané podle ekvalizovaných histogramů vypočtených v jednotlivých regionech [8]. Ve skutečnosti počítáme vážený průměr hodnot v rozích mřížky. Obrázek (4.10) můžeme použít i pro lineární interpolaci.

Lineární interpolace

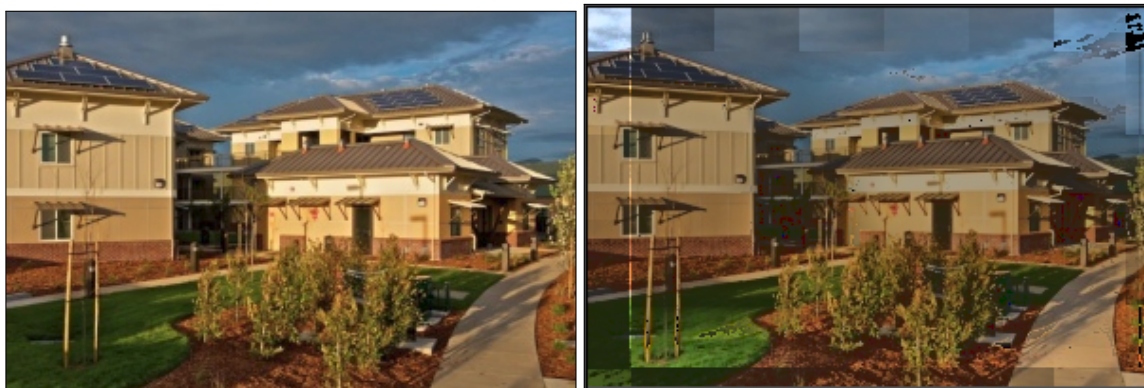
Pomocí lineární interpolace, ze které mimo jiné vychází i bilineární interpolace (4.2.3), se získá hodnota pixelu ležící mezi dvěma pixely mající známou hodnotu. Hodnota bodu s' se určí za pomoci rovnice:

$$s' = (1 - x)g_A(s) + xg_B(s) \quad (4.20)$$

v případě, že bod leží v horní či dolní zelené oblasti nebo pomocí rovnice:

$$s' = (1 - y)g_A(s) + yg_C(s) \quad (4.21)$$

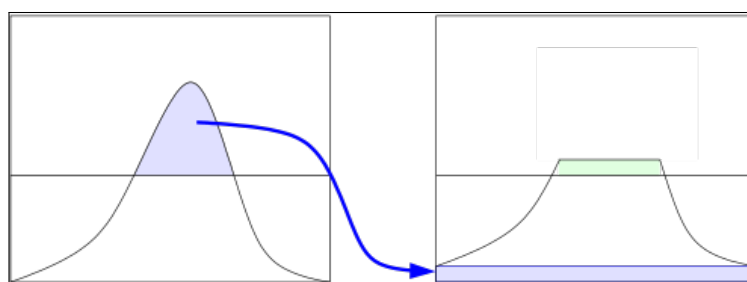
v případě, že bod leží v levé či pravé zelené oblasti.



Obrázek 4.11: Původní barevný obrázek a transformovaný obrázek s interpolacemi

4.2.4 Clip Limit

Samotný Adaptive Histogram Equalization nejen že zvýší detaily v obraze, ale bohužel také zvýší bílý šum, což je u některých případů nežádoucí. Proto se zrodil nápad spočívající v nastavení maximálního počtu četnosti v každé třídě histogramu pro zamezení zvýšení kontrastu bílého šumu. K tomu se právě používá clip limit. Ten spočívá v tom, že nalezne průměrnou hodnotu tříd histogramu a ořízne jí podle parametru na požadovanou velikost. Pixels, které převyšují maximální povolenou výšku histogramu se ale musí rovnoměrně rozložit mezi zbývajících třídy histogramu, aby zůstala zachována celková četnost všech tříd histogramu, která značí počet pixelů. Ořezaný histogram ale nemusí korespondovat se vstupním obrázkem, proto není CLAHE příliš vhodná technika pro obrazy využívající se ve výzkumných odvětvích. U diskrétního histogramu se ořezané pixely nemusí rozložit rovnoměrně, proto se rovnoměrně rozdělí tolik pixelů, kolik to jen jde a zbývajících převyšujících pixely se rozdělí pouze mezi některé třídy, nejlépe do těch, ze kterých byly odříznuty pro zachování co nejvíce podobného obrázku s originálem[8] [14].



Obrázek 4.12: Ukázka clipnutí histogramu. Zdroj obrázku [2]

Následující algoritmus popisuje použití clip limitu při oříznutí histogramu, konkrétně jeho implementaci do projektu.

Data: *inputHist, cl*

Result: *clippedHist*

//nastavení nejvyšší četnosti v jednom binu

int *maxBinValueCL* = (floor)(*averageVAlue*cl*);

int *toLayoutBins* = 0;

//sečtení všech prvků přesahující nastavenou četnost, vytvoření základu pro clipnutý histogram

for *int i* = 0; *i* < 256; *i*++ **do**

if *inputHist[i]* > *maxBinValueCL* **then**

toLayoutBins += *inputHist[i]* - *maxBinValueCL*; *clippedHist[i]* +=
 maxBinValueCL;

end

end

//uniformní rozdělení přesahujících binů do clipnutého histogramu, v případě větší četnosti prvků, než je povolená hodnota je povolená hodnota odečtena

int *toLayoutForBin* = *toLayoutBins* / 256;

int *restBins* = *toLayoutBins* - *toLayoutForBin**256;

for *int i* = 0; *i* < 256; *i*++ **do**

clippedHist[i] += *toLayoutForBin*;

end

// rozdělení zbývajících binů

for *int i* = 0; *i* < *restBins*; *i*++ **do**

clippedHist[i]++;

end

Algoritmus 3: Clip limit histogramu



Obrázek 4.13: Ukázkové obrázky použití různých úrovní clip limitu, horní levý obrázek - původní, horní prostřední obrázek 0.25, horní pravý obrázek 0.5, dolní levý obrázek 0.75, dolní pravý obrázek 1

Kapitola 5

Porovnání metod

V této kapitole se porovnávají vybrané vlastnosti implementovaných metod. U metod se porovnávaly tyto nejhlavnější vlastnosti:

- Rychlost
- Paměťová náročnost
- Efektivita
- Použití

I když nebyly metody porovnávány podrobněji pro získání konkrétnějších údajů z jejich porovnání, z hlediska jejich zcela odlišného zpracování může svůj účel splnit pouze objektivní porovnání.

5.1 Rychlost

Z hlediska rychlosti je jednoznačně rychlejší metoda CLAHE, protože nepoužívá tak složité operace, jako je optimalizace (4.8) a (4.10) u Art Photographic Detail Enhancement. Rychlost optimalizace se enormně sníží přidáním i jednoho parametru, protože se stavový prostor pro vyhledávání optimálních hodnot rozšíří o jednu dimenzi. Rychlost ovšem může záležet i na způsobu implementace a použití knihoven pro optimalizaci. Ačkoli některé knihovny pro optimalizaci byly poměrně rychlé, tak vracely velmi nepřesné výsledky (převážně z důvodu použití stochastických metod či pravděpodobnostních optimalizačních metodách, kterými je například knihovna Pallas.

5.2 Paměťová náročnost

Paměťová náročnost je u obou metod velice vysoká, ovšem protože Art Photographic Detail Enhancement využívá ke zvýraznění detailů také optimalizaci, tak je (především u většího rozměru obrázku) značně náročnější na paměť z důvodu optimalizace, k jejímuž provedení je potřeba matice o enormních rozměrech (jedná se o čtvercovou matici, jejíž rozměry se rovnají počtu pixelů v obrázku (4.1.3)). CLAHE oproti tomu ukládá do paměti pouze histogramy regionů, které jsou co se týče rozměrů, konstantní (podle počtu tříd v histogramu) a paměť v tomto případě ovlivňuje pouze počet regionů.

5.3 Efektivita

Metoda Art Photographic Detail Enhancement je, co se týče efektivity lepší, protože oproti metodě CLAHE, která zvýrazňuje detaily v oblasti regionu, je první zmíněná metoda vyhledává a zlepšuje na úrovni jednotlivých pixelů (4.1.2). Metoda je také založena na složitějších procedurách (optimalizace) pro co největší zvýraznění detailů než metoda druhá, která pouze vyrovnává histogram.

5.4 Použití

Obě metody se mohou s největší pravděpodobností použít víceméně ve stejných oblastech. Ovšem je nutno dodat, že i když obě metody zvýrazňují detaily na poměrně vysoké úrovni, tak se může stát, že metody některé detaily překreslí. Z tohoto důvodu se metoda Art Photographic Detail Enhancement hodí spíše na převádění fotografií na umělecké (3.1.3), kdežto metoda CLAHE se dá do jisté míry použít u rentgenových snímků pro jejich lepší čtení či zvýraznění snímků s tmavými či světlými oblastmi (3.2.3).

Kapitola 6

Závěr

Cílem bakalářské práce bylo porozumět náročnějším metodám, které se zabývají zvýrazněním detailů v obraze a následnými implementacemi. První z těchto metod byla založena především na více průchodech potřebných pro zpracování obrazu a také na použití optimalizačních funkcí (3.1). Druhá, značně jednodušší metoda byla založena na modifikované ekvalizaci histogramu (3.2). I když byla první metoda zajímavější, protože se zabývala hlubší detekcí detailů a jejich zvýrazněním více než metoda druhá, je nutné podotknout, že realizace procedur této metody, které byly založené na optimalizaci, byly značně složitější k naimplementování v programovacím jazyce C++, protože bylo potřeba nalézt správnou knihovnu pro optimalizaci implementovanou v tomto jazyce, která je rychlá, nenáročná na paměť a umožňuje vložit funkci k optimalizaci takovým způsobem, jakým je žádáno. Také stojí za zmínku, že původní autoři tuto metodu implementovali v MATLAB, který je právě pro tyto složité matematické operace lepší z hlediska efektivity a rychlosti, kterými MATLAB disponuje. Takže se nabízí otázka, v jakém programovacím jazyce by měly být podobné metody implementovány. Podle mého názoru by měly být tyto metody implementovány převážně v MATLABu, protože ten je na tyto operace s maticemi uzpůsoben lépe než C++ a především je méně náročný na paměť a mnohonásobně rychlejší.

Literatura

- [1] *A C++ implementation of Image Smoothing via L0 Gradient Minimization* .
- [2] *Adaptive histogram equalization*.
URL https://en.wikipedia.org/wiki/Adaptive_histogram_equalization
- [3] *CGAL 4.12 - Linear and Quadratic Programming Solver* .
URL https://doc.cgal.org/latest/QP_solver/index.html
- [4] *CGAL - The Computation Geometry Algorithms Library* .
- [5] *Dark*.
URL <https://www.imdb.com/title/tt5753856/>
- [6] *OpenCV* .
- [7] *Positive-definite matrix* .
- [8] Heckbert, P. S.: *Graphics GEMS IV*.
URL <https://doc.lagout.org/Others/Game%20Development/Programming/Graphics%20Gems%204.pdf>
- [9] Hoshang, K.: *Real-Time Sky Color with Effect of Sun's Position*.
URL https://www.researchgate.net/figure/CIE-color-space-for-xy-chromaticitys_fig2_236592295?_sg=oI_grNMyLXYs2KgeKbL10w1RnkD5PskmKgDNqonFHYVXMybCef6NyT7oprBG-LxCJaLgeJUNJqRm1iN4fwE7Rw
- [10] Hruz, M.: *Adaptive Histogram Equalization, Histogram processing and Convolution*.
URL http://www.kky.zcu.cz/uploads/courses/mpv/01/lesson01_materials.pdf
- [11] Lemire, D.: *Should computer scientists keep the Lena picture?*
URL <https://i.stack.imgur.com/o1z7p.jpg>
- [12] Macková, K.: *Řešení úlohy kvadratického programování pomocí metod vnitřních bodů* .
- [13] Pelikán, J.: *Filtrace Obrazu* .
- [14] Pizer, S. M.: *Adaptive Histogram Equalization and Its Variations* .
- [15] Son, M. e. a.: *Art-Photographic Detail Enhancement*.
URL http://www.cs.ums1.edu/~kang/Papers/kang_eg14.pdf
- [16] Son, M. e. a.: *Art-Photographic Detail Enhancement - More Results*.
URL http://cg.postech.ac.kr/research/art_photograph/results.php

- [17] Szelinski, R.: *Computer Vision, Algorithms and Applications* .
- [18] Xu, L. e. a.: *Image Smoothing via L_0 Gradient Minimization*.
URL <https://sites.fas.harvard.edu/~cs278/papers/l0.pdf>
- [19] Čadík, M.: *Perceptual Evaluation of Color-to-Grayscale Image Conversions*.
URL http://cadik.posvete.cz/color_to_gray_evaluation/

Seznam obrázků

2.1	Lena - Původní obrázek. Zdroj obrázku [11]	4
2.2	Příklady použití šumu a jejich potlačení. (nahore potlačení Gaussovkého šumu Gaussovým filtrem, dole potlačení "náhodného" šumu mediánovým filtrem)	6
2.3	Obrázek před a po ekvalizaci histogramu.	7
3.1	Ukázkový obrázek po použití metody. Zdroj obrázku [16]	8
3.2	Ukázka metody CLAHE. Zdroj obrázku [19]	10
3.3	Detail nežádoucího artefaktu obrázku po ekvalizaci histogramu, pokud má region v místě artefaktu převážně stejnou hodnotu pixelu. Výřez z vlastního obrázku.	11
3.4	Ukázka ekvalizace obrázku s mlžným oparem v lese. Původní obrázek převzat z [5]	11
4.1	Původní obrázek (nahore) a obrázek po maximalizaci detailů (dole). Obrázek převzat z [16]	18
4.2	Vstupní barevný obrázek. Zdroj obrázku nenalezen.	19
4.3	Vstupní obrázek ve stupních šedi. Zdroj obrázku [10]	19
4.4	Ukázka barevného modelu Yxy. Zdroj obrázku [9]	20
4.5	Vstupní barevný obrázek (vlevo) a výsledný obrázek po ekvalizaci	23
4.6	Vstupní obrázek v úrovních šedi (vlevo) a výsledný obrázek po ekvalizaci	23
4.7	Původní barevný obrázek a transformovaný obrázek bez interpolací	24
4.8	Původní obrázek ve stupních šedi a transformovaný obrázek bez interpolací	24
4.9	Mapa regionů obrázku a jeho významné body. Zdroj obrázku [2]	25
4.10	Bilineární interpolace. Zdroj obrázku [8]	26
4.11	Původní barevný obrázek a transformovaný obrázek s interpolacemi	27
4.12	Ukázka clipnutí histogramu. Zdroj obrázku [2]	27
4.13	Ukázkové obrázky použití různých úrovní clip limitu, horní levý obrázek - původní, horní prostřední obrázek 0.25, horní pravý obrázek 0.5, dolní levý obrázek 0.75, dolní pravý obrázek 1	29

Příloha A

Obsah přiloženého média

- `readme.txt` - textový soubor obsahující uživatelskou příručku a návod ke kompilaci
- `bp.pdf` - tisknutelná podoba bakalářské práce
- `src/` - složka se zdrojovými soubory metod
- `images/` - složka s obrázky pro vyzkoušení
- `latex/` - složka se zdrojovými kódy latexu pro vysázení bakalářské práce

Příloha B

Použité technologie a knihovny

V této kapitole jsou popsány technologie a knihovny, které byly při implementaci metod použity. Tyto technologie a knihovny jsou použité především pro práci a manipulaci s obrazem (OpenCV) nebo využívají různé druhy programování pro optimalizaci funkcí za účelem získání takových hodnot pro neznámé proměnné, za kterých bude účelová funkce dosahovat minimální (nebo maximální) hodnoty. Mezi tyto druhy programování se řadí například lineární programování, nelineární programování nebo kvadratické programování. Podrobnější popis, jak tyto metody fungují, jsou zobrazeny v následujících podsekcích téhle kapitoly.

B.1 OpenCV knihovna

Open Source Computer Vision (zkráceně OpenCV) je, jak už název napovídá, open source multiplatformní knihovna pro manipulaci s obrazovými daty. Tato knihovna bylo původně vyvinuta společností Intel. Tato knihovna kromě zpracování obrazu dokáže i rozeznávat objekty v obraze, zachytit pohyb ve videu nebo do jisté míry i strojové učení. Knihovnu využívají jazyky C, C++ nebo Python [6].

B.2 CGAL knihovna

CGAL je softwarový projekt, který poskytuje snadný přístup k efektivním a spolehlivým geometrickým algoritmům ve formě knihovny C++. CGAL se používá v různých oblastech, které potřebují geometrické výpočty, jako jsou geografické informační systémy, počítačem podporovaný design, molekulární biologie, lékařské zobrazování, počítačová grafika a robotika.

Knihovna nabízí datové struktury a algoritmy jako triangulace, Voronovy diagramy, booleovské operace na polygonech a polyhedrách, zpracování bodových souborů, uspořádání křivek, generování povrchových a objemových ok, zpracování geometrie, alfa tvary, konvexní algoritmy trupu, tvarová analýza, AABB a KD stromy...[4]

Z knihoven byla použita knihovna, která řeší kvadratické programování při maximalizaci detailů (4.1.3).